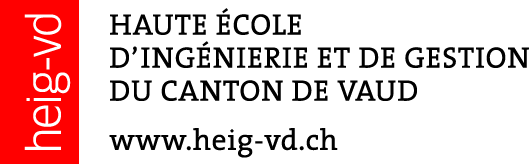
June 2017

Laboratory Report

HTTP Infrastructure

RES

Prof. Olivier Liechti

Bichon Passuello

Max Caduff

Step 1:

The objective of this first part is to implement a static HTTP Apache server, using existing Docker images.

We used the Docker image « PHP » (<https://hub.docker.com/_/php/>) as a base, and the template « Stylish Portfolio » from startbootstrap (<https://startbootstrap.com/template-overviews/stylish-portfolio/>) slightly modified as the content to be displayed. The website is located in the content/ folder, which is copied in the var/www/html/ folder of the Container.

The DockerFile to do this is set as follows:

808

FROM php:7.0-apache

COPY content/ var/www/html/

To build the image, we run « sudo docker build -t res/apache\_php . » from the folder containing

content/ and the Dockerfile. (apache-php-image).

Then, to run the container, we should give it the local port to communicate with it (here 9090) and the port of the container on which it should listen (here 80). Hence the command: « sudo docker run -p 9090:80 res/apache\_php »

In order to interact with the machine, we can launch a bash instance when running it with:

« sudo docker run -i -p 9090:80 res/apache\_php /bin/bash »

we can then explore the filesystem and find the location of the Apache config files, which can be found under:

/var/lib/apache2/conf

Step 2:

In this part we needed to implement a dynamic server in parallel to the Static server from Step 1, based on a node.js image to generate the response, and express.js to manage the server behavior.

The server generates a fake Finnish dialog made of a random number of random sentences pairs, the first is a 5-word question, the answer has an unbounded number of words.

The files to run the server are located under src/, they are copied in the /opt/app folder of the vm, and they are run from there.

The DockerFile is as follows:

FROM node:6.10

COPY src /opt/app

CMD ["node", « opt/app/index.js »]

The command to build the image is: « sudo docker build -t res/express\_students . »

The command to run the container is: « sudo docker run -p 9090:3000 res/express\_students »

We need once again to map the correct ports of the machine, since the container is listening on his port 3000.

Step 3 :

Still using the PHP 7.0 image as a base, the reverse proxy virtual hosts are configured as follow :

The 001-reverse-proxy.conf file is « hardcoded » with the IP adresses automatically attributed to the containers when launched. Those are found by first running a bash on the server-hosting containers with « docker exec -it <container-name> /bin/bash » and then « docker inspect <container-name> | grep -i ipadd ».

001-reverse-proxy.conf :

VirtualHost \*:80>

ServerName demo.res.ch

#ErrorLog ${APACHE\_LOG\_DIR}/error.log

#CustomLog ${APACHE\_LOG\_DIR}/access.log combined

ProxyPass "/api/finnish/" "http://172.17.0.3:3000/"

ProxyPassReverse "/api/finnish/" "http://172.17.0.3:3000/"

ProxyPass "/" "http://172.17.0.4:80/"

ProxyPassReverse "/" "http://172.17.0.4:80/"

</VirtualHost>

Any request to the url /api/finnish/ will be redirected to the express\_dynamic vm, and any GET / request will be redirected to the apache\_static vm.

To run the static server : « sudo docker run -d --name apache\_static res/apache\_php »

The attributed ip address is :

"IPAddress": "172.17.0.4",

To run the dynamic server : « sudo docker run -d --name express\_dynamic res/express\_students »

The attributed ip address is :

"IPAddress": "172.17.0.3".

Step 4 :

We needed to add the line « 0.0.0.0 [demo.res.ch](http://demo.res.ch/) » to the hosts file in /private/etc/ because the proxy only responds to its name.

172.17.0.2 is the ip attributed to apache\_static vhost

172.17.0.3 is the ip attributed to express\_dynamic vhost

Added command to DockerFile to update apt-get and install vim in every DockerFile:

« RUN apt-get update && apt-get install -y vim »

The js command result replaces the text in the « about » section from the static page by a finnish sentence taken from the dynamic server every 2 seconds. The requests and answers are made via the container running the proxy.

This line was added at the end of the body of the index.html file from the static image:

<script src ="js/finnish.js"></script>

and the corresponding code was created in the js/ folder from the same image:

$(function() {

function loadFinnish () {

$.getJSON( "/api/finnish/",

function (finnish) {

console.log(finnish);

var message = "No one is speaking";

if (finnish.length > 0) {

message = finnish[0].req;

}

$(".about").text(message);

});

};

loadFinnish();

setInterval( loadFinnish, 2000);

});

Everything else is the same as in the last step.

Step 5:

The goal of this step was to create a way to give the ip adresses of the static and dynamic servers to the proxy without having to rebuild the image each time.

The solution is based on environment variables, which are set with the needed ip when launching the proxy (with -e ENV=value ), and then uses php to run a script which gets the value of those variables and writes them in the configuration file from step 3.

The DockerFile is modified as follows:

FROM php:7.0-apache

RUN apt-get update && apt-get install -y vim

COPY apache2-foreground /usr/local/bin/

COPY conf/ /etc/apache2/

COPY templates/ /var/apache2/templates/

RUN a2enmod proxy proxy\_http

RUN a2ensite 000-\* 001-\*

RUN service apache2 restart

the script is similar to the 001-reverse-proxy.conf file, except that we added the code to get the environment variables and put them instead of the fixed IPs:

<?php

$ip\_stat = getenv('STATIC\_APP');

$ip\_dyna = getenv('DYNAMIC\_APP');

?>

<VirtualHost \*:80>

ServerName demo.res.ch

#ErrorLog ${APACHE\_LOG\_DIR}/error.log

#CustomLog ${APACHE\_LOG\_DIR}/access.log combined

ProxyPass '/api/finnish/' 'http://<?php print "$ip\_dyna"?>/api/finnish/'

ProxyPassReverse '/api/finnish/' 'http://<?php print "$ip\_dyna"?>/api/finnish/'

ProxyPass '/' 'http://<?php print "$ip\_stat"?>/'

ProxyPassReverse '/' 'http://<?php print "$ip\_stat"?>/'

</VirtualHost>

to run the php script, we customized a file from the existing php image, apache2-foreground, which is luckily run when starting the container. We added a few lines, only the last one starting by « php … »  is useful to run the script.

lines added:

#RES features

echo "static address received $STATIC\_APP"

echo "dynamic address received $DYNAMIC\_APP"

php /var/apache2/templates/config-template.php > /etc/apache2/sites-available/001-reverse-proxy.conf

The command to launch the proxy is now: (here we launched the dynamic server before the static one to invert their ip adresses)

docker run -d -e STATIC\_APP=172.17.0.3:80 -e DYNAMIC\_APP=172.17.0.2:3000 -p 8080:80 res/apache-rp